

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Staš Voler

Napredna spletna aplikacija za naročanje hrane

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Igor Rožanc

Ljubljana 2015

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi predstavite razvoj napredne spletne aplikacije za naročanje hrane preko spleta. Poleg sistematične predstavitve razvoja in ustrezne izvedbe tehničnega dela aplikacije naj bo glavni poudarek naloge na vgraditvi mehanizmov za uporabniku prilagojeno ponudbo hrane. V ta namen ločeno obravnavajte različne skupine uporabnikov, nalogo pa zaključite s primerjavo s podobnimi obstoječimi spletnimi stranmi.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Staš Vler sem avtor diplomskega dela z naslovom:

Napredna spletna aplikacija za naročanje hrane

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Igorja Rožanca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 20. januarja 2015

Podpis avtorja:

Zahvaljujem se vsem svojim bližnjim, ki so me podpirali in spodbujali v času študija. Posebno se zahvaljujem mentorju, viš. pred. dr. Igorju Rožancu, za koristne nasvete in pomoč pri izdelavi mojega diplomskega dela.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Uporabljene tehnologije in orodja	3
2.1	Uporabljene tehnologije	3
2.1.1	HTML in CSS	3
2.1.2	PHP	4
2.1.3	MySQL	5
2.1.4	JavaScript	5
2.1.5	AJAX	5
2.2	Uporabljene knjižnice in ogrodja	6
2.2.1	Knjižnici jQuery in jQuery UI	6
2.2.2	Ogrodje Bootstrap	6
2.2.3	Druge knjižnice in ogrodja	7
2.3	Orodja	7
2.3.1	Notepad++	7
2.3.2	MySQL Workbench	7
2.3.3	Orodja za razvijalce Firefox in Chrome	8
3	Razvoj spletne aplikacije	9
3.1	Diagrami primerov uporabe	9

KAZALO

3.2	Načrtovanje podatkovne baze	15
3.3	Implementacija	18
3.3.1	Komunikacija s strežnikom	18
3.3.2	Zaledni sistem	19
3.3.2.1	Pregledovanje aktivnih naročil	20
3.3.2.2	Samodejno razvrščanje pic v kategorije	22
3.3.3	Prednja stran	27
3.3.3.1	Prikaz restavracije in menija	28
3.3.3.2	Iskanje restavracije	29
3.3.3.3	Košarica	30
3.3.3.4	Pregled naročila	31
3.3.4	Sistem za predlaganje pic	32
3.3.5	Odzivna spletna stran	34
4	Analiza in odziv uporabnikov	35
4.1	Primerjava z obstoječimi aplikacijami	35
4.2	Pravilnost samodejnega razvrščanja pic	37
4.3	Odziv uporabnikov	37
5	Sklepne ugotovitve	39

Seznam uporabljenih kratic

kratica	angleško	slovensko
HTML	Hyper Text Markup Language	označevalni jezik za nadbesedila
CSS	Cascading Style Sheets	kaskadne slogovne podloge
PHP	Hypertext Preprocessor	programski jezik za razvoj dinamičnih spletnih strani
JSON	JavaScript Object Notation	JavaScript objekt za prenašanje podatkov
XML	Extensible Markup Language	razširljiv označevalni jezik
AJAX	Asynchroneus JavaScript and XML	Asinhroni JavaScript in XML
AJAJ	Asynchroneus JavaScript and JSON	Asinhroni JavaScript in JSON
SQL	Structured Query Language	strukturirani povpraševalni jezik
HTTP	Hyper Text Transfer Protocol	protokol za prenos nadbesedil
DOM	Document Object Model	objekt z vsebino spletnega dokumenta
SVG	Scalable Vector Graphics	skalabilna vektorska grafika

Povzetek

Cilj diplomske naloge je bil izdelati spletno aplikacijo za naročanje hrane (oziroma pic) prek spleta. Glavni poudarek naloge je omogočiti čimbolj učinkovito prodajo. V ta namen je bil izdelan poseben mehanizem za samodejno razvrščanje pic v več kategorij, ki skupaj s podatki o uporabnikovih preteklih nakupih omogoča, da se uporabniku predlaga nabor najprimernejših pic.

Prvi del diplomske naloge opisuje uporabljene tehnologije in orodja. Pri razvoju aplikacije je uporabljen programski jezik PHP in druge spletne tehnologije (HTML, CSS, JavaScript, AJAX in SQL) ter knjižnice in ogrodja (jQuery, Twitter Bootstrap in druge). Za shranjevanje podatkov je uporabljena podatkovna baza MySQL.

V drugem delu je opisan razvoj spletne aplikacije od ideje, načrtovanja in izdelave do uporabe končnega izdelka. Spletno mesto sestavljata prednji in zadnji del aplikacije, ki omogoča uporabo različnim tipom uporabnikov. Zasnovano je tako, da se prilagaja na širino zaslona naprave. Podrobno je opisan predvsem algoritem za razvrščanje in predlaganje pic, ki je glavna odlika aplikacije.

V zadnjem delu je predstavljena primerjava s podobnimi spletnimi stranmi in analiza uporabe, v zaključku pa še pregled opravljenega dela, glavne težave pri razvoju ter ideje za nadaljnje delo.

Ključne besede: spletna aplikacija, spletno naročanje hrane, PHP, MySQL, avtomatsko predlaganje.

Abstract

The purpose of the thesis was to produce a web application for ordering food (pizzas) online. The thesis mainly focuses on providing the most effective sales possible. For that reason we built a special mechanism for automatic categorisation of pizzas in different categories, which combined with data on a user's past purchases provided a selection of the most appropriate pizzas for that specific user.

The first part of the thesis describes the technologies and tools used. In developing the application PHP scripting language and other online technologies were used (HTML, CSS, JavaScript, AJAX and SQL), as well as libraries and tools (jQuery, Twitter Bootstrap and others). MySQL database was used for data storage.

The second part describes the development of the web application from the idea, design and production to the use of the final product. The website is composed of a fore-end and back-end part of the web application, which enables different types of users to use the website. It is built in such a way that it is responsive to different screen widths. The thesis in detail describes the algorithm for pizza categorisations and suggestions, which is the main advantage of the application.

The final part provides a comparison of the website with similar websites, an analysis of the application and concludes with an overview of the work done, the main problem in development and the idea for future work.

Keywords: web application, online food ordering, PHP, MySQL, automatic suggestions.

Poglavje 1

Uvod

V zadnjih letih se je z razvojem tehnologij in vsesplošno dostopnostjo do spleta povečalo število različnih spletnih trgovin, med njimi tudi spletne aplikacije, ki so namenjene izključno naročanju hrane. Te so pravi razcvet doživele prav z razvojem pametnih telefonov in tabličnih računalnikov, saj nam omogočajo dostop do spleta prav na vsakem koraku.

Tako kot v svetu, se tudi pri nas iz leta v leto povečuje število tovrstnih trgovin. Nekatere restavracije se odločijo za svoj lasten sistem za naročanje (Julči¹ ali Paparotti²), v večini primerov pa se odločajo za priključitev k že izdelanim aplikacijam, ki ponujajo naročanje hrane iz različnih restavracij (Ehrana³, Lakomlacen⁴, Halohrana⁵).

Med pregledovanjem spletnih strani, ki ponujajo naročilo hrane iz različnih restavracij, smo opazili, da se dejansko nobena ne osredotoča na prodajo posamezne vrste hrane, npr. na pice, domačo hrano, mehiško ali kitajsko, ampak so vse bolj ali manj splošne.

Zato je bil naš cilj v okviru naloge izdelati spletno aplikacijo, ki bi se osredotočala predvsem na prodajo ene vrste hrane na čimbolj učinkovit način.

¹<http://www.julci.si>

²<http://www.paparotti-2.si>

³<http://www.ehrana.si>

⁴<http://www.lakomlacen.si>

⁵<http://www.halohrana.si>

Odločili smo se, da izdelamo aplikacijo, ki se osredotoča na prodajo pic. Pica nam ponuja ogromno možnosti za sestavo, saj jo lahko obložimo s številnimi sestavinami. Uporabniku poleg naročanja že vnaprej sestavljenih pic omogoča tudi sestavo lastne pice. Kar pa je najpomembnejše: aplikacija na podlagi predhodnih naročil uporabniku predlaga pice, ki bi kar najbolj ustrezale njegovemu okusu, in ga tako pritegne h končnemu nakupu.

Spletna stran je zasnovana tako, da se oblika prilagaja glede na širino zaslona naprave (odzivni pogled). S tem dosežemo, da je stran enako pregledna na mobilni napravi kot na osebni računalniku.

Aplikacija je v grobem razdeljena na dva dela, in sicer na zaledni del, ki je namenjen administratorjem, lastnikom restavracij in njihovim zaposlenim za sprejemanje naročil ter urejanje podatkov, in prednji del, ki je namenjen vsem uporabnikom za naročanje hrane.

V prvem delu se bomo na kratko dotaknili uporabljenih tehnologij, orodij in knjižnic, ki smo jih uporabili pri razvoju spletne aplikacije. Nato pa se bomo osredotočili na razvoj in kratko predstavitev delovanja aplikacije.

Poglavje 2

Uporabljene tehnologije in orodja

Za razvoj spletne aplikacije smo za komunikacijo s strežnikom in dinamičnost uporabili programska jezika PHP in JavaScript, za postavitev gradnikov ter videz pa HTML in CSS. Poleg tega smo uporabili različne spletne tehnologije in knjižnice, ki so opisane v podpoglavjih. Za hranjenje podatkov smo uporabili podatkovno bazo MySQL.

Za razvoj aplikacije smo uporabili tekstovni urejevalnik Notepad++, za izdelavo podatkovne baze pa MySQL Workbench. Pri razvoju smo si pomagali še s sodobnimi spletnimi brskalniki (Firefox in Chrome), ki imajo vgrajena orodja za razvijalce.

2.1 Uporabljene tehnologije

2.1.1 HTML in CSS

HTML [1] je standardiziran označevalni jezik za izdelavo spletnih strani. Z njim določamo strukturo spletnega dokumenta. Z uporabo značk postavljamo različne elemente, ki jih nato brskalnik interpretira in izriše na zaslon. Začetki razvoja jezika HTML segajo v leto 1991, ko je bilo definiranih prvih 18 elementov, med njimi tudi značka za naslov, povezavo in sezname.

Zanimivo je, da je kar polovico prvotnih elementov obstalo tudi v različici HTML 4, ki je bila definirana leta 1999. Zadnja različica je HTML5, ki nam med drugim omogoča gradnike za predvajanje multimedijskih vsebin, risanje dvodimenzionalnih grafičnih vsebin in celo lokalno shranjevanje podatkov.

S pomočjo CSS-ja [2, 3] določamo stile posameznih ali skupin elementov HTML. Stil določa, kako se bo element prikazoval v brskalniku. Elementom lahko določamo različne lastnosti, kot so velikost, obroba, odmik, barva itd. Na spletno stran jih največkrat vključujemo v ločenih dokumentih CSS, lahko pa jih zapisujemo neposredno v dokument HTML. Razvit je bil zaradi potrebe po tem, da se ločijo elementi HTML in vsebina od same predstavitve strani. S tem se poveča preglednost vsebine in bistveno prepreči ponavljanje kode.

2.1.2 PHP

PHP [4, 5] je objektno orientiran programski jezik, ki se izvaja na strežniški strani in se uporablja za razvoj dinamičnih spletnih strani. Začetki segajo v leto 1995, ko je bila izdana prva različica. Napisan je v programskem jeziku C, kar se opazi tudi pri sami sintaksi. Gre za izjemno priljubljen jezik, najbrž prav zaradi svoje preprostosti. Po nekaterih raziskavah naj bi se PHP uporabljal pri več kot tretjini vseh spletnih strani. [6]

Za delovanje ima PHP na strežniški strani vzpostavljen sistem za razčlenjevanje in interpretiranje kode. Odjemalec s pomočjo protokola HTTP pošlje zahtevek na strežnik, tam pa interpreter PHP kodo interpretira in jo izvede. Ko se koda izvede, jo strežnik pošlje nazaj odjemalcu kot odgovor v HTML, XML, JSON ali kakšni drugi obliki.

PHP podpira povezave do večine današnjih najpogostejše uporabljenih podatkovnih baz, kot so MySQL, IBM DB2, MSSQL, Oracle in SQLite. [7]

2.1.3 MySQL

MySQL [8, 9] je sistem za upravljanje z relacijskimi podatkovnimi bazami (SUPB). Relacijsko podatkovno bazo si lahko predstavljamo kot množico med seboj povezanih dvodimenzionalnih tabel, kjer so shranjeni podatki. Za manipulacijo s podatki uporablja jezik SQL, ki je najbolj razširjen standardiziran povpraševalen jezik za dostopanje do podatkovnih zbirk. MySQL je drugi najpopularnejši SUPB. [10] Je izjemno zmogljiv sistem, ki lahko hrani ogromne količine podatkov in je hiter, če so le podatki ustrezno strukturirani. Nenazadnje uporablja MySQL za nekatere svoje storitve tudi Google, Facebook in Twitter.

2.1.4 JavaScript

JavaScript [11, 12] je objektno-skriptni programski jezik za izdelavo interaktivnih spletnih strani. Izvaja se na strani odjemalca. Program običajno vključujemo v dokument HTML prek ločenih datotek JavaScript ali pa ga pišemo tja kar neposredno. JavaScript lahko uporabljamo za shranjevanje podatkov, dinamično spreminjanje stilov ali vrednosti elementov, izračune itd.

2.1.5 AJAX

AJAX [13, 14] je skupek povezanih spletnih tehnologij za razvoj interaktivnih spletnih strani. Glavni namen je komunikacija med odjemalcem in strežnikom, ki se izvaja asinhrono v ozadju. To pomeni, da lahko spletna aplikacija prejema ali pošilja podatke, ne da bi bilo potrebno stran ponovno osvežiti. S tem dosežemo, da se med odjemalcem in strežnikom prenaša bistveno manj podatkov, tako da je spletni strežnik manj obremenjen. Poleg tega se aplikacija bistveno hitreje odziva na spremembe in s tem izboljša uporabniško izkušnjo.

Čeprav se v kratici AJAX skriva XML ta ni nujno obvezen za izmenjavo podatkov. Danes se za izmenjavo podatkov zelo pogosto uporablja JSON.

Posledica tega je, da se je pojavil novi izraz za asinhrono komunikacijo, in sicer AJAX.

2.2 Uporabljene knjižnice in ogrodja

Za hitrejši razvoj smo pri implementaciji uporabili nekatere odprtokodne knjižnice in orodja. Največ smo si pomagali s knjižnicama jQuery oz. jQuery UI ter z ogrodjem Twitter Bootstrap za izdelavo odzivnega pogleda.

2.2.1 Knjižnici jQuery in jQuery UI

jQuery [15] je odprtokodna knjižnica JavaScript za hitrejšo in lažje pisanje kode JavaScript. Po nekaterih raziskavah gre za daleč najpogostejše uporabljeno knjižnico JavaScript in je postala skoraj nepogrešljiva za slehernega spletnega razvijalca. [16] Zelo praktična in enostavna je pri uporabi klicev AJAX, potrebne je bistveno manj kode kot pri uporabi golega JavaScripta, posledica tega pa je preglednejša koda.

jQueryUI [17] je skupek grafičnih kontrolnih elementov, namenjenih za izboljšanje grafičnega uporabniškega vmesnika. Nudi nam številne interakcije z elementi (povleci in spusti, spreminjanje velikosti elementov) na uporabniku prijazen način, vizualne gradnike (drsniki oz. drsni gumbi, izbiranje datuma, samodokončanje besed) in različne efekte (barvne animacije, animirano skrievanje in odkrivanje elementov).

2.2.2 Ogrodje Bootstrap

Izdelava odzivnih spletnih strani je lahko zelo dolgotrajna, zato so se pojavila številna ogrodja, da bi to delo olajšala. Eden izmed najbolj razširjenih je ogrodje Bootstrap [18], ki je plod Twitterjevih razvijalcev. Bootstrap temelji na uporabi treh spletnih tehnologij, in sicer HTML, CSS in JavaScript, ki poskrbijo, da se elementi prilagajajo glede na širino zaslona naprave. Poleg

enostavnih elementov so izdelane še številne sestavljene komponente, kot so navigacijski meniji, spustni meniji, različne tabele in številčenje strani.

2.2.3 Druge knjižnice in ogrodja

Pri razvoju smo uporabili še številne druge knjižnice in ogrodja, ki so pripomogle predvsem k izboljšanju uporabniškega vmesnika. Od teh jih večina za svoje delovanje uporablja prav knjižnico jQuery. Nekatere izmed njih so Select2 (nadgradnja osnovnega elementa select) [19], Star Rating (ocenjevanje) [20], FancyBox (prikaz slik) [21], Bootstrap WYSIHTML5 (pisanje obogatenih besedil) [22], Font Awesome (za ikone) [23] in Raphaël (prikaz vektorskih grafik) [24].

2.3 Orodja

2.3.1 Notepad++

Notepad++ [25] je preprost tekstovni urejevalnik za pisanje oz. urejanje izvirne kode. Napisan je v C++ in nudi podporo za ogromno število programskih jezikov. Na prvi pogled je videti kot zelo enostavno orodje, vendar ponuja celo kopico uporabnih funkcij, npr. samodokončanje besed, barvanje sintakse, skrivanje in odkrivanje določenih delov kode, primerjanje dveh datotek in še veliko drugih funkcij. Vgrajen ima tudi brskalnik FTP, tako da se lahko povežemo na strežnik in urejamo datoteke, kot da bi jih imeli shranjene na lastnem disku.

2.3.2 MySQL Workbench

MySQL Workbench [26] je izjemno zmogljivo orodje za delo s podatkovnimi bazami. Uporabili smo ga predvsem zaradi hitrega in enostavnega načrtovanja podatkovnega modela. Orodje omogoča pretvorbo podatkovnega modela v podatkovno bazo. Za nas generira skripte SQL z vsemi pripadajočimi relacijami, njihovimi lastnostmi in povezavami. Sam program

ponuja še bistveno več, od urejevalnika za izvajanje poizvedb SQL, urejanje podatkov v tabelah, do orodja za analizo učinkovitosti podatkovnih baz.

2.3.3 Orodja za razvijalce Firefox in Chrome

Brskalnika Firefox in Chrome imata kot drugi sodobni brskalniki vgrajena orodja za razvijalce spletnih aplikacij. Ta orodja nam ponujajo razhroščevalnik kode JavaScript, pregledovalnik sloga in elementov DOM, urejanje sloga, pregled omrežnega delovanja in virov. Orodje nam omogoča izbiranje elementov na strani, tako da nam posameznih elementov ni nujno treba iskati v pregledovalniku DOM, ampak to storimo samo z enim klikom na strani. Ponuja tudi možnost odzivnega pogleda, kjer lahko spreminjamo širino in višino zaslona, tako da imamo pogled obnašanja strani na različnih velikostih zaslonov.

Poglavje 3

Razvoj spletne aplikacije

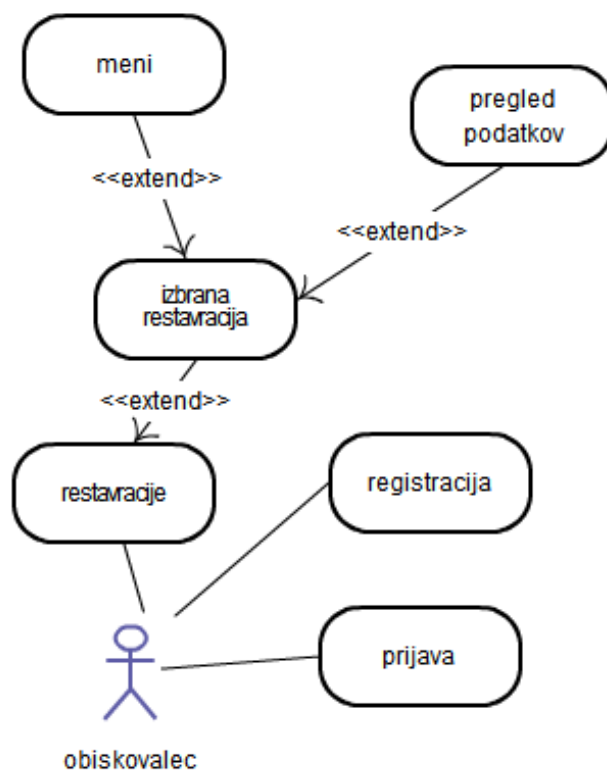
Razvoj spletne aplikacije je bil razdeljen na štiri dele. Najprej smo podrobno preučili problem in naredili krajšo analizo zahtev. Iz nje smo nato izdelali diagrame primerov uporabe, s pomočjo katerih smo kasneje načrtovali podatkovno bazo. Na koncu je sledila še implementacija.

3.1 Diagrami primerov uporabe

Iz analize zahtev smo za boljši pregled nad vsemi funkcionalnostmi izdelali diagrame primerov uporabe, ki vključujejo vse aktivnosti in igralce. Igralci so v našem primeru uporabniške skupine, in sicer *obiskovalec*, *uporabnik*, *zaposleni*, *lastnik restavracije* in *administrator*. Vsaka izmed teh skupin ima dovoljenja oz. omejitve, katere aktivnosti lahko izvaja.

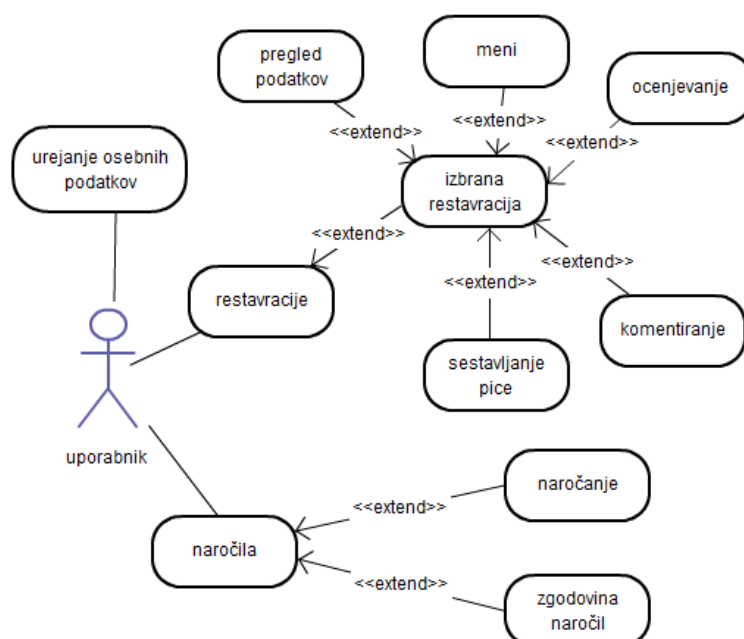
Naša spletna aplikacija je logično razdeljena na prednji in zaledni del. Prednji del predstavlja glavni del aplikacije: namenjen je obiskovalcem in registriranim uporabnikom za pregledovanje restavracij, njihove ponudbe ter naročanje. Zaledni del pa je namenjen lastnikom restavracij in njihovim zaposlenim ter administratorjem za pregledovanje in urejanje podatkov.

V nadaljevanju so natančneje predstavljeni diagrami za vsakega igralca posebej.



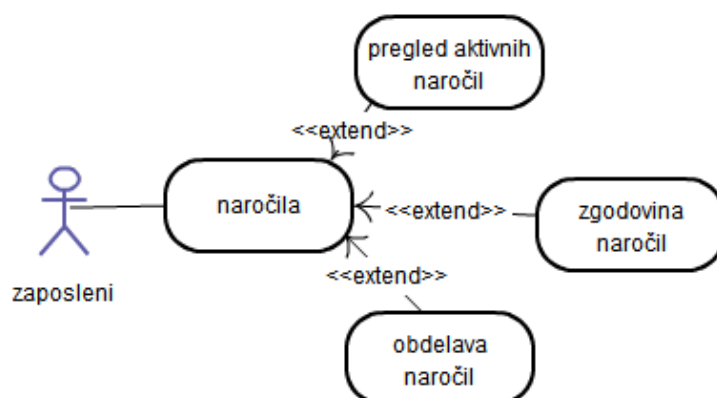
Slika 3.1: Diagram primerov uporabe za obiskovalca. Prikaže aktivnosti prednjega dela aplikacije.

Naključnemu obiskovalcu je omogočen pregled restavracij in pregledovanje ponudb, ne more pa naročiti hrane in biti aktiven pri komentiranju ter ocenjevanju restavracij. Za pridobitev teh pravic sta najprej potrebna registracija in prijava v sistem. Registracija v sistem je zelo enostavna: obiskovalec mora vpisati svoje ime in priimek ter elektronski naslov in geslo, ki ju bo uporabljal za prijavo. Po uspešni registraciji uporabnik prejme na svoj elektronski naslov potrditev o registraciji in povezavo do aktivacije računa. Ko je račun aktiviran, se lahko obiskovalec prijavi v sistem.



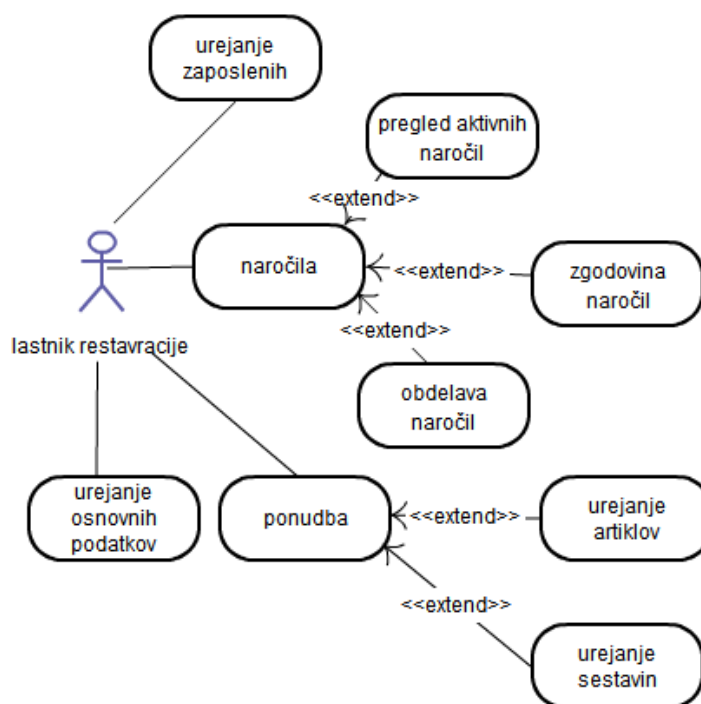
Slika 3.2: Diagram primerov uporabe za uporabnika. Prikazuje aktivnosti prednjega dela aplikacije.

Po uspešni registraciji in prijavi obiskovalca v sistem so mu dodeljene pravice uporabnika. To so pravice za izvajanje naročil in pregledovanje njihove zgodovine. Poleg tega ima tudi možnost urejanja osebnih podatkov in naslovov, ki jih lahko uporablja pri naročanju hrane z dostavo. Vsak uporabnik ima lahko shranjenih več naslovov za dostavo, enega pa lahko označi kot privzetega, tako da mu ni treba ob vsakem naročanju napisati naslova. Aktivno lahko sodeluje tudi pri ocenjevanju in komentiranju restavracij.



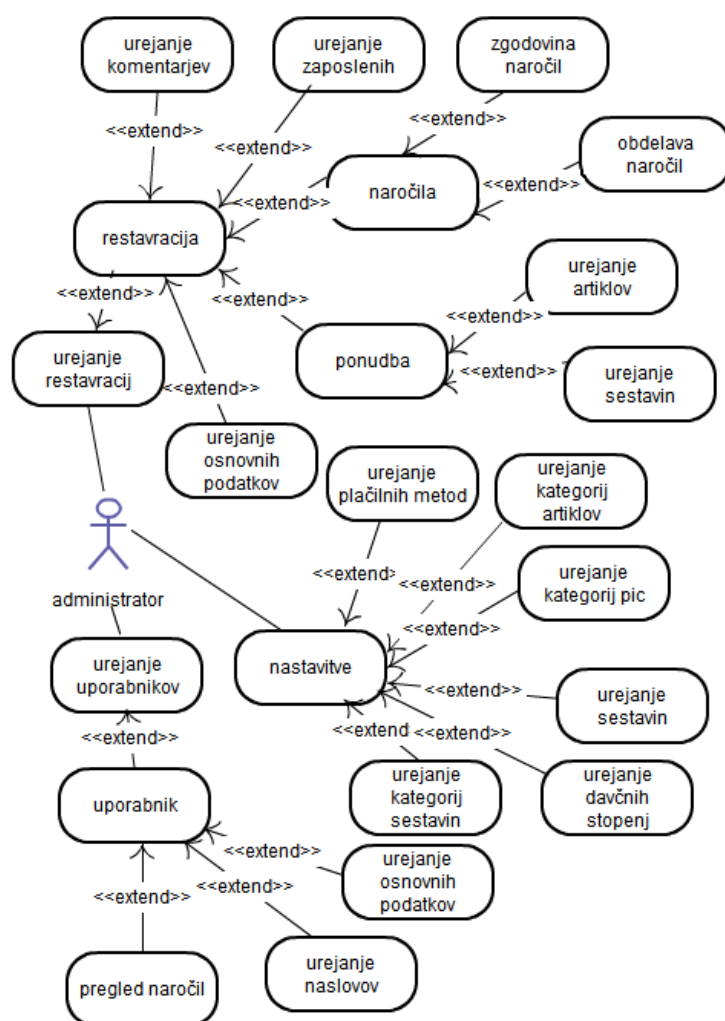
Slika 3.3: Diagram primerov uporabe za zaposlenega. Prikazuje aktivnosti zalednega dela aplikacije.

Vsaka restavracija ima lahko neomejeno število zaposlenih. Zaposleni ima pristojnosti za pregledovanje in upravljanje z naročili. Gre predvsem za spreminjanje njihovih statusov (naročilo oddano, prejeto, prevzeto, zaključeno in zavrnjeno). Poleg nastavljanja statusov lahko pri vsakem naročilu napiše komentar za naročnika (npr. ob zavrnitvi naročila poda razlog za zavrnitev).



Slika 3.4: Diagram primerov uporabe za lastnika restavracije. Prikazuje aktivnosti zalednega dela aplikacije.

Lastnik restavracije ima tako kot zaposleni poleg možnosti pregledovanja in urejanja naročil še dovoljenja za urejanje osnovnih podatkov o restavraciji, zaposlenih in seveda urejanje ponudbe. Na preprost način lahko v svoji meni dodaja ali odstranjuje artikle ali pa samo spreminja podatke, kot npr. ceno, opis, fotografijo itd. Določeni artikel lahko začasno tudi odstrani iz ponudbe, tako da ga preprosto deaktivira. Lastnik ureja tudi sestavine, določa privzete cene, ki so pri posameznih artiklih ponujane kot dodatki. Če restavracija omogoča naročanje sestavljenih pic, določa tudi sestavine, med katerimi uporabnik lahko izbira.



Slika 3.5: Diagram primerov uporabe za administratorja. Prikazuje aktivnosti zalednega dela aplikacije.

Največ dovoljenj ima zagotovo administrator, ki nadzira praktično celoten del aplikacije. V sistem dodaja in odstranjuje restavracije, manipulira s podatki posamezne restavracije, pregleduje njihova naročila itd. Poleg restavracij ureja še ostale uporabnike, jih dodaja ali odstranjuje, ureja njihove osnovne podatke in naslove ter pregleduje njihova naročila.

Z diagrama je razvidno, da lahko upravlja tudi z nekaterimi nastavitvami.

Administrator določa kategorije (*pice, solate, pijače, jedi z žara . . .*), v katere lahko lastnik restavracije uvršča svoje artikle. Vsaki kategoriji določi, kateri atributi so obvezni in kateri opcijski pri urejanju posameznega artikla. Na primer, kategorija pica ima nastavljeno obvezno polje za določanje njihovih sestavin, medtem ko je lahko pri kakšni drugi kategoriji to zgolj neobvezno polje ali pa ga sploh ni.

Eden najpomembnejših delov pa je urejanje kategorij pic in urejanje sestavin. Aplikacija namreč skrbi za samodejno razvrščanje pic v ustrezne kategorije. Tu administrator določa pravila za ustrezno razvrščanje pic na podlagi njihovih sestavin.

Administrator ureja še davčne stopnje, po katerih so obdavčeni posamezni artikli, in plačilne metode, med katerimi lahko restavracije izbirajo za načine plačila.

3.2 Načrtovanje podatkovne baze

Na podlagi izdelanih diagramov primerov uporabe smo se lotili načrtovanja podatkovne baze. Naša baza vsebuje 32 povezanih relacij oz. tabel. Po njihovih funkcijah bi jo lahko razdelili na šest logičnih enot, in sicer: *naslov*, *uporabnik*, *restavracija*, *sestavina*, *artikel* in *naročilo*.

- **Logično enoto *naslov* sestavljajo tri tabele:**

- **address** (*naslov*),
- **post** (*pošta*) in
- **region** (*regija*).

V tabeli **naslov** hranimo podatke o naslovih uporabnikov in restavracij, v tabeli **pošta** in **regija** pa so vnaprej shranjene pošte ter regije v Sloveniji. Vsaka pošta pripada natanko eni regiji.

- Logično enoto *uporabnik* sestavljajo tri tabele:

- `user` (`uporabnik`),
- `user_group` (`uporabnik_skupina`) in
- `user_address` (`uporabnik_naslov`).

V tabeli `uporabnik_skupina` so vnaprej določene uporabniške skupine. Vsak uporabnik pripada eni uporabniški skupini.

V tabeli `uporabnik_naslov` so zabeleženi vsi naslovi posameznega uporabnika, saj aplikacija omogoča, da ima uporabnik shranjenih več naslovov.

- Logično enoto *restavracija* sestavlja devet tabel:

- `restaurant` (`restavracija`),
- `restaurant_payment_method` (`restavracija_plačilna_metoda`),
- `comment_restaurant` (`komentar_restavracija`),
- `rate_restaurant` (`ocena_restavracija`),
- `employee_restaurant` (`zaposleni_restavracija`),
- `opening_hours_restaurant` (`odpiralni_čas_restavracija`),
- `opening_hours` (`odpiralni_čas`),
- `restaurant_delivery_posts` (`restavracija_dostava_pošte`) in
- `custom_ingredients` (`sestavine_po_meri`).

Vsaki restavraciji lahko določimo odpiralni čas in čas dostave, za kar poskrbita tabeli `odpiralni_čas` in `odpiralni_čas_restavracija`. Zaposlene beležimo v `zaposleni_restavracija`, komentarje in ocene restavracij pa v `komentar_restavracija` in `ocena_restavracija`. Vsaka restavracija lahko za posamezne sestavine določi njihovo ceno (kot dodatne sestavine pri naročanju) in če omogoča sestavljanje pic, lahko določa, katere sestavine so na voljo in katere ne. Te podatke hranimo

v tabeli `sestavine_po_meri`. V tabeli `restavracija_dostava_pošte` hranimo poštne številke krajev, kamor restavracija dostavlja hrano.

- Logično enoto *sestavina* sestavljajo tri tabele:

- `ingredient` (`sestavina`),
- `ingredient_category` (`sestavina_kategorija`) in
- `ingredient_categories` (`sestavina_kategorije`).

Vsaki sestavini pripada natanko ena glavna kategorija, lahko pa ji določimo tudi podkategorije.

- Logično enoto *artikel* sestavlja deset tabel:

- `article` (`artikel`),
- `article_category` (`artikel_kategorija`),
- `article_ingredients` (`artikel_sestavine`),
- `article_size` (`artikel_velikost`),
- `pizza_categories` (`pica_kategorije`),
- `pizza_category` (`pica_kategorija`),
- `additional_ingredients` (`dodatne_sestavine`),
- `possible_additional_ingredients` (`možne_dodatne_sestavine`),
- `pizza_ingredient_perecentage` (`pica_delež_sestavlin`) in
- `vat_rate` (`ddv_stopnja`).

Vsak artikel spada v eno splošno kategorijo. Vsak artikel, ki spada v splošno kategorijo pica, pa uvrstimo še v eno glavno kategorijo pic in v poljubno število podkategorij. V tabeli `pica_delež_sestavlin` shranjujemo neke vrste pravila za avtomatizirano razvrščanje pic v kategorije. V tabeli `možne_dodatne_sestavine` beležimo sestavine, ki jih restavracija nudi kot dodatke posameznim jedem. Tabelo `dodatne_sestavine` bi lahko uvrstili tudi v enoto naročilo, beleži pa sestavine, ki jih izbere kupec kot dodatek posamezni jedi.

- Logično enoto *naročilo* sestavljajo štiri tabele:

- `order` (`naročilo`),
- `order_article` (`naročilo_artikel`),
- `order_status` (`naročilo_status`) in
- `payment_method` (`plačilna_metoda`).

V vsakem naročilu je lahko več artiklov, ki jih beležimo v `naročilo_artikel`. Naročilo ima status in podatek o plačilni metodi, po kateri bo naročilo plačano.

3.3 Implementacija

Po izdelavi načrta in podatkovne baze smo se lotili implementacije. Glavno vodilo je bilo izdelati spletno aplikacijo, ki bo pregledna, hitra in lahka za uporabo. Najprej smo se lotili izdelave zalednega dela, nato pa še prednjega. V tem podpoglavju bodo razloženi najbolj zanimivi deli aplikacije.

3.3.1 Komunikacija s strežnikom

Ker gre za tip aplikacije, ki veliko komunicira in si izmenjuje podatke s strežnikom, smo izdelali posebno funkcijo JavaScript, s pomočjo knjižnice jQuery, ki skrbi za klice AJAX.

Funkcija `ajaxCall` sprejme tri parametre. S prvim podamo url do skripte PHP na strežniku, drugi je za podatke, ki jih pošiljamo strežniku, s tretjim pa določimo, ali gre zgolj za znakovne tipe podatkov ali pa prenašamo tudi datoteke.

Podatke, ki jih podamo kot parameter, tipično pridobimo iz spletnega obrazca in jih shranimo v objekt `FormData` kot množico parov *ključ/vrednost*.

Za prenos datotek moramo parameter `contentType` v funkciji jQuery `ajax` nastaviti na *false*. S tem dosežemo, da se `Content-Type` v glavi zahtevka definira sam, kajti sicer pride kasneje do težav. Prav tako nastavimo

parameter `processData` na *false*, s čimer preprečimo, da bi jQuery podatke pretvarjal v znakovni niz.

```
function ajaxCall(url, data, multipart){

    if(multipart){
        var myContentType = false;
        var myProcessData = false;
    }else{
        var myContentType =
            'application/x-www-form-urlencoded; charset=UTF-8';
        var myProcessData = true;
    }

    $.ajax({
        type: "POST",
        url: "/ajax/" + url,
        dataType: 'json',
        contentType: myContentType,
        processData: myProcessData,
        data: data
    }).done(function(msg) {
        ...
    });
    return false;
}
```

3.3.2 Zaledni sistem

Zaledni sistem nam je zaradi številnih funkcionalnosti vzel kar precejšen del časa za razvoj. Pri implementiranju smo morali biti še posebno pazljivi pri omejitvah pravic. V ta namen smo izdelali sistem, ki preverja, za kateri tip uporabnika gre in če ima dovoljenje za prikaz določenega dela aplikacije. Če dovoljenja nima, ga preusmerimo na drugo stran ali pa napišemo ustrezno obvestilo.

Na sliki 3.6 je prikazan tipičen pogled na razporeditev elementov v zalednem delu. Na vrhu je meni z glavnimi kategorijami, sledi naslov in pod njim orientacijski meni ali drobtine (*ang. breadcrumb*). V glavnem delu strani je na levi strani navigacijski meni, desno pa glavno okno, kjer so tipično postavljeni spletni obrazci.

Najbolj zanimivi deli zalednega dela se zagotovo skrivajo med nastavitvami, ki jih upravlja administrator, kamor spada tudi sistem za razvrščanje pic v ustrezne kategorije. Za lastnike in njihove zaposlene pa je verjetno najbolj aktualno pregledovanje aktivnih naročil.

Slika 3.6: Prikaz urejanja podatkov posamezne restavracije v zalednem sistemu za administratorje.

3.3.2.1 Pregledovanje aktivnih naročil

Za uspešno obratovanje restavracije je ključnega pomena čim hitrejša odzivnost, zato smo nekaj več časa posvetili ravno pregledovanju aktivnih naročil.

S slike 3.7 je razvidno, da se naročila osvežujejo samodejno na določen časovni interval, kar je ponazorjeno tudi grafično z vrstico napredka. Če pride do novega ali več novih naročil, se na zaslonu prikaže opozorilo, da so prišla nova naročila, prav tako dobimo opozorilo na oknu brskalnika oz. zavihka, kjer je sicer izpisan naslov strani.

Naročila so zaradi preglednosti izpisana v razširljivi tabeli in urejena glede na čas. S pritiskom na gumb '+' na skrajni desni strani v vrstici tabele

lahko lastnik ali zaposleni odpirajo in zapirajo podrobnosti naročila. Tam so izpisani vsi potrebni podatki o naročilu, vsi artikli in količina, način plačila, čas dostave in komentar naročnika.

Lastniki in zaposleni spreminjajo status naročila na enostaven način prek spustnega menija. Status se odraža tudi pri barvi vrstice posameznega naročila. Novo prejeta naročila so obarvana rumeno, prejeta in prevzeta modro, zaključena zeleno ter zavrnjena rdeče.

Komentar pa podajo tako, da napišejo besedilo v ustrezno polje in pritisnejo na gumb 'Shrani komentar'.

Aktivna naročila

(1) Novo naročilo!

Osvežite čez 34 s

Čas naročila	Naročnik	Naslov	Znesek	Status
5. 12. 2014 13.51 (8 minut nazaj) TAKOJ	Staš Voler	Levčeva ulica 17 1234 Mengeš	6,50 EUR (Gotovina)	naročilo oddano
Staš Voler Levčeva ulica 17 1234 Mengeš 031 846 520		Količina Artikel Cena Znesek 1x Kmečka mala (pice) 6,50 6,50	Za plačilo EUR 6,50 Gotovina 6,50	
		Čas naročila	5. 12. 2014 13.51 (8 minut nazaj)	
		Čas dostave	TAKOJ	
		Komentar restavracije:	<div> </div> <div>Shrani komentar</div>	
5. 12. 2014 9.33 (266 minut nazaj) 8. 12. 2014 14.00 (čez 4321 minut)	Staš Voler	Levčeva ulica 17 1234 Mengeš	12,50 EUR (Gotovina)	naročilo prejeto

Slika 3.7: Prikaz pregledovanja aktivnih naročil za lastnike restavracij in njihove zaposlene.

3.3.2.2 Samodejno razvrščanje pic v kategorije

Zaradi bolj učinkovitega sistema za predlaganje pic, ki je bolje razložen v kasnejšem poglavju, smo se odločili za izdelavo mehanizma za samodejno razvrščanje pic v ustrezne kategorije. Posamezno pico bi sicer lahko določili ročno po lastni presoji, vendar bi tako lahko prišlo do nekonsistentnosti, saj bi bila uvrstitev v kategorijo lahko preveč subjektivna. Poleg tega aplikacija omogoča sestavljanje pice in tudi te bi si želeli razporediti v določene kategorije.

V ta namen smo definirali pravila, po katerih se bodo pice razvrščale. To smo storili tako, da smo posamezni kategoriji pic (sirove, mesne, morske, zelenjavne ...) določili deleže sestavin, po katerih se kasneje naredi izračun za uvrstitev v najprimernejšo glavno kategorijo.

The image shows a web interface for setting ingredient proportions for three categories of pizza. Each category has a slider and two input fields.

Kategorija	Slučajna vrednost (%)	Max št. sestavin	Max pomembnost sestavin
Sirovi	1 - 100%	0	0
Meso	0 - 0%	0	0
Zelenjava	0 - 10%	1	100

Slika 3.8: Prikaz določanja deležev sestavin za kategorijo sirovih pic.

S slike 3.8 je razvidno, da posamezni kategoriji sestavin preprosto z drsniki določimo odstotkovno zgornjo in spodnjo mejo. Pri tem veljajo pravila glede določanja zgornje in spodnje meje, ki so ključnega pomena pri računanju, in sicer:

Delež 0–100

Pomeni, da je lahko prisotno poljubno število sestavin, ki pripada tej kategoriji. Pri računanju se te kategorije ne upošteva.

Delež 1–100

Pomeni, da gre za glavno in hkrati obvezno kategorijo sestavin ter je pri računanju najbolj pomembna.

Delež 0–0

Pomeni, da se v posamezni kategoriji pice ne sme pojaviti sestavina s posamezne kategorije sestavin.

Ostali deleži

Pri računanju se upoštevata zgornja in spodnja meja. Če je delež teh sestavin pod spodnjo ali nad zgornjo mejo, naredimo pri izračunu nekakšno redukcijo.

Omeniti je še treba, da imajo tudi posamezne sestavine določen odstotek pomembnosti vpliva na izračun, kajti vseh ne moremo enačiti med seboj. Tipičen primer je oliva, ki je sicer zelenjava, vendar pa se pri računanju ne more primerjati npr. s papriko, saj se v nekaterih restavracijah oliva pojavi skoraj na vsaki pici.

V nadaljevanju je po korakih prikazan potek izračuna za uvrščanje pice v glavno kategorijo in nato še v podkategorije. To so:

1. Najprej pridobimo vse kategorije in podkategorije sestavin ter pri vsaki zabeležimo število sestavin, ki pripada tej kategoriji. Poleg zabeležimo še pomembnost teh sestavin.
2. Nato za vsako kategorijo sestavin izračunamo njihove deleže na podlagi števila sestavin in pomembnosti.

3. Ko pridobimo deleže sestavin za vsako kategorijo, začnemo z računanjem najprimernejše kategorije s pomočjo pravil, ki določajo spodnjo in zgornjo mejo. Za vsako kategorijo pic računamo t. i. statusne točke. Statusne točke so dejansko neko število, ki ponazarjajo primernost za uvrstitev pice v kategorijo. Na začetku so statusne točke nastavljene na 0, potem pa jih višamo oz. nižamo. Izračun statusnih točk pa naredimo po naslednjem postopku:
- (a) Če je pri kategoriji sestavine določeno pravilo *Delež 1–100* in je delež teh sestavin višji od 0 %, potem statusne točke povišamo tako, da pomnožimo delež sestavin v kategoriji s 100. V primeru, da je delež teh sestavin 0 %, potem statusne točke znižamo za 100 in s statusno zastavico označimo, da gre za t. i. neprimerno kategorijo.
 - (b) Če je pri kategoriji sestavine določeno pravilo *Delež 0–0* in je delež sestavin 0 %, potem statusne točke povišamo za 50. V nasprotnem primeru jih znižamo za 50 in s statusno zastavico označimo, da gre za neprimerno kategorijo.
 - (c) Če gre pri kategoriji za pravilo *Ostalih deležev*, potem pogledamo, če je delež znotraj mej ali ni. Če je znotraj mej, statusne točke povišamo tako, da delež pomnožimo z 10, sicer pa znižamo za delež pomnožen z 10. Pri tem delež pomnožimo še s koeficientom, ki predstavlja odmik od zgornje oz. spodnje meje.
Pri pravilu *Ostalih deležev* velja tudi pravilo največje število dovoljenih sestavin in največji dovoljeni delež sestavin. Če so pogoji za posamezno pravilo izpolnjeni, potem statusne točke povišamo za 10, sicer jih za 10 znižamo.
4. Po izračunu statusnih točk pregledamo, katera kategorija pic nima zastavice za t. i. neprimerno kategorijo in med njimi poiščemo tisto z največ statusnimi točkami. Če so med njimi vse kategorije neprimerne, potem upoštevamo samo število statusnih točk.

5. Podkategorije določimo bistveno bolj preprosto kot glavno. Vsaki sestavini lahko namreč določimo podkategorije, na drugi strani pa imamo podkategorije pic, pri katerih je določeno, katere podkategorije sestavin morajo vsebovati. Tako samo pogledamo, katere podkategorije se ujemajo.

Primer samodejnega razvrščanja

Na primer, da imamo glavne kategorije z določenimi pravili: *sirove*, *mesne*, *zelenjavne*, *vegetarijanske*, *morske* in *sadne*. Imamo pa tudi podkategoriji: *pikantna* in *kisla*.

V ustrezne kategorije želimo uvrstiti pico z naslednjimi sestavinami: *peleti*, *sir*, *paprika*, *cebula*, *tuna*, *origano*.

Najprej pridobimo vse kategorije sestavin, ki se upoštevajo pri računanju, in pri vsaki zabeležimo število pripadajočih sestavin ter pripišemo pomembnost. S pomočjo teh podatkov izračunamo še deleže posameznih sestavin.

Glavna kategorija	Število sestavin	Pomembnost	Delež sestavin $\left(\frac{\text{Št. sestavin v kat.} * \text{Pomembnost}}{\text{Št. vseh sestavin} * 100} \right)$
Siri	1	100	$(1/4) * (100/100) = 0.25$
Meso	0	0	0
Zelenjava	2	200	$(2/4) * (200/100) = 1$
Morsko	1	100	$(1/4) * (100/100) = 0.25$
Gobe	0	0	0
Sadje	0	0	0
Podkategorija			
Pikantna	0	-	-
Kisla	0	-	-

Tabela 3.1: Razvrstitev sestavin v kategorije in podkategorije ter izračun njihovih deležev.

Nato s pomočjo pravil za razvrščanje izračunamo statusne točke za vsako kategorijo pic in označimo t. i. neprimerne kategorije.

Kategorija pic	Statusne točke	Primerna/ Neprimerna
Sirove	$100 \cdot 0.25 + 50 - 10 \cdot (1 + 0.9)$ $-10 - 10 - 50 + 50 = 36$	N
Mesne	$10 \cdot (0.25 + 0) + 10 + 10 - 100 - 10 \cdot (1 + 24.5)$ $+10 + 10 - 50 + 50 = -312.5$	N
Zelenjavne	$-10 \cdot (0.25 + 2.2125) + 10 + 10 + 10 \cdot (0 + 0)$ $+10 + 10 + (100 \cdot 1) - 50 + 50 = 115.375$	N
Vegetarijanske	$-10 \cdot (0.25 + 3.95) + 10 + 10 + 50$ $+ (100 \cdot 1) - 50 + 50 = 128$	N
Morske	$10 \cdot (0.25 + 0) + 10 + 10 + (10 \cdot 0 + 0) + 10 + 10$ $-10 \cdot (1 + 8.7) + 10 + 10 + (100 \cdot 0.25) + 50 = 40.5$	P ✓
Sadne	$10 \cdot (0.25 + 0) + 10 + 10 + (10 \cdot 0 + 0) + 10 + 10$ $-(10 \cdot (1 + 6)) + 10 + 10 - 50 - 100 = -157.5$	N

Tabela 3.2: Izračun statusnih točk za vsako kategorijo pic.

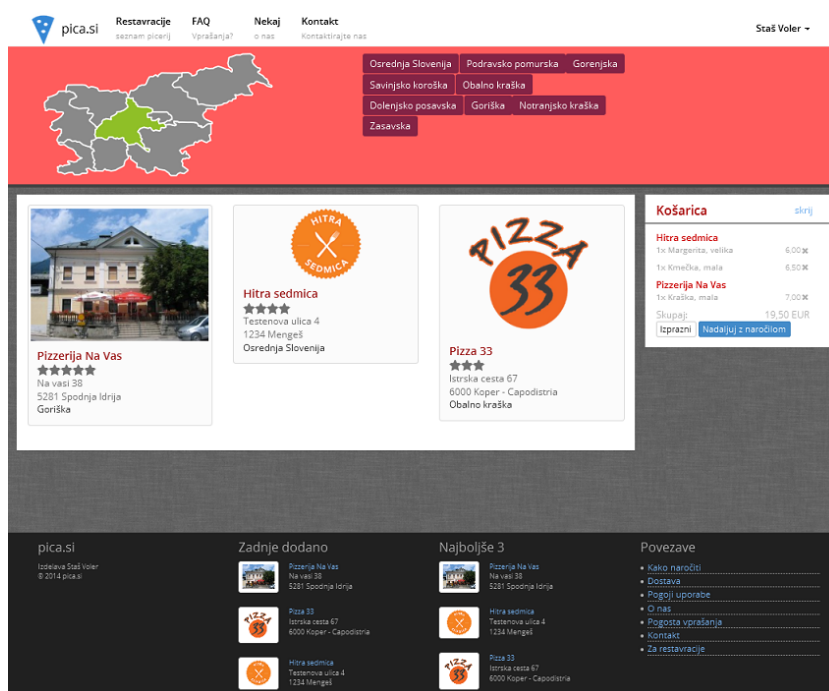
Ko izračunamo statusne točke izmed vseh primernih kategorij poiščemo tisto, ki ima največ točk. V našem primeru samo morska kategorija nima zastavice za neprimerno kategorijo, tako da podano pico uvrstimo med morske.

Na koncu še pregledamo, če morda kakšna sestavina pripada določeni podkategoriji. V našem primeru ni takšne sestavine, zato je ne uvrstimo nikamor. Če bi vsebovala npr. *feferone*, pa bi jo uvrstili med pikantne.

3.3.3 Prednja stran

Prednja stran je vidna vsem obiskovalcem in uporabnikom ter predstavlja glavni del aplikacije. Prek tega sistema se uporabnik registrira in prijavi v sistem, išče ter pregleduje restavracije, opravlja naročila, pregleduje zgodovino naročil in ureja osebne podatke.

Prednja stran je glede na postavitev elementov logično razdeljena na tri dele, in sicer na glavo, telo ter nogo. Glava je razdeljena na dva dela: v zgornjem delu se nahaja meni, v spodnjem pa zemljevid, razdeljen na regije, in gumbi posameznih regij. Zemljevid smo izdelali z uporabo skalabilne vektorske grafike SVG, za prikaz pa smo uporabili knjižnico Raphaël. Telo je razdeljeno na levi in desni del: na levem delu je tipično glavna vsebina strani, na desnem pa nakupovalna košarica. V nogi se nahajajo osnovni podatki o strani, tri najboljše ocenjene restavracije, nazadnje dodane restavracije in povezave do različnih podstrani.



Slika 3.9: Prikaz prednje strani.

3.3.3.1 Prikaz restavracije in menija

Vsaka restavracija se prikazuje na svoji strani, tako kot je na sliki 3.10. Zgornji del je rezerviran za najosnovnejše podatke o restavraciji. Tam se nahaja logotip, naziv restavracije, naslov in koristni podatki za naročnika. Z različnimi simboli je označeno, ali restavracija nudi dostavo oz. prevzem, možnosti plačila ... Na primer: če gremo z miško preko določenega simbola, se pojavi napis, ki ta simbol podrobneje opisuje. Če gremo preko simbola, na katerem je kartica, se izpiše seznam plačilnih kartic, s katerimi lahko plačujemo. V zgornjem delu je še z zvezdicami ponazorjena ocena restavracije, kjer jo lahko uporabniki tudi sami ocenijo. V spodnjem delu so v zavihku *O restavraciji* še nekateri drugi podatki (spletna stran, elektronski naslov ...), prikaz lokacije na zemljevidu in grafično obogaten odpiralni čas (če je restavracija odprta, je polje posameznega dneva obarvano zeleno, sicer pa rdeče). V zavihku *Komentarji* so izpisani vsi komentarji in za uporabnike možnost dodajanja le-teh.



Hitra sedmica
 ☉ (odprto še 10 h in 41 min)
 Testenova ulica 4
 1234 Mengeš



 (na podlagi 1 ocene)

Min. naročilo: 5,00 EUR
 Cena dostave: 1,00 EUR
 Čas dostave: 30-45 min
 Sestavljene pica: 12,30 EUR

O restavraciji

Komentarji

Osnovni podatki
 Testenova ulica 4
 1234 Mengeš
 info@h7.si
 http://www.h7.si
 070 772 772
 Kje smo?


Odpiralni čas

Restavracija	Dostava
Ponedeljek	9.00 - 15.00
Torek	9.00 - 22.00
Sreda	9.00 - 22.00
Četrtek	9.00 - 22.00
Petek	9.00 - 22.00
Sobota	Zaprto
Nedelja	Zaprto

Slika 3.10: Prikaz posamezne restavracije.

Meni, prikazan na sliki 3.11, je sestavljen iz filtra, kjer izbiramo, katere vrste jedi se nam prikazujejo, in prikaza artiklov. Vsak artikel ima sliko, ime, opis, ceno in gumb za dodajanje v košarico. Nekateri artikli imajo lahko več cen (tipičen primer je pica, kjer ponavadi izbiramo med različnimi velikostmi) oz. možnost izbiranja dodatkov. Dodatki se nam prikažejo s pritiskom na gumb '+ dodatne sestavine'. Dodatne sestavine so razporejene po kategorijah, tako da imamo nad njimi malce boljši pregled. Prikazana cena nad gumbom se samodejno spreminja glede na izbrane lastnosti in dodatke.



Slika 3.11: Prikaz menija pri posamezni restavraciji.

3.3.3.2 Iskanje restavracije

Da bi bilo iskanje restavracije čim hitrejše in čimbolj enostavno, smo izdelali filter, s katerim lahko zelo hitro najdemo zelene restavracije. Za izbiranje regij lahko uporabimo kar zemljevid oz. gumbe v glavi strani ali pa jo izberemo s pomočjo spustnega menija v filtru. V drugem spustnem meniju pa lahko izbiramo med kraji, ki se nahajajo v izbrani regiji. Poleg izbiranja regij in krajev lahko iskanje omejimo še tako, da prikazujemo samo tiste, ki so trenutno odprte, ali pa tiste, ki nudijo plačilo na študentske bone, imajo dostavo oz. osebni prevzem ter ponujajo sestavljanje pice.

Spletna aplikacija si s pomočjo piškotkov zapomni našo nazadnje iskano regijo in kraj, tako da aplikacija naslednjič samodejno ponudi restavracije, ki smo jih nazadnje iskali.



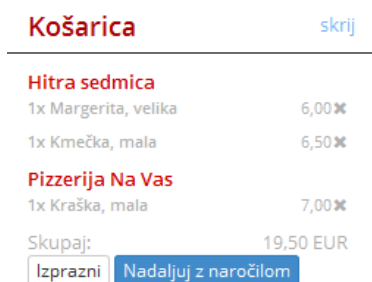
Slika 3.12: Prikaz filtra za hitrejšo iskanje restavracij.

3.3.3.3 Košarica

Spletna nakupovalna košarica je eden izmed najpomembnejših elementov pri naročanju. Nahaja se v zgornjem desnem kotu strani in je vseskozi vidna na zaslonu. Omogoča naročanje hrane prek ene ali več restavracij hkrati.

Košarico smo implementirali z jezikom JavaScript, z uporabo knjižnice jQuery. Izdelali smo dva razreda jQuery: **Article** in **Cart**. Razred **Article** je namenjen beleženju in manipuliranju vseh ključnih podatkov posameznega artikla. To so podatki o ceni, velikosti, količini, dodatnih sestavinah itd. Razred **Cart** pa je namenjen dodajanju, brisanju in urejenemu shranjevanju artiklov (gre za objekte razreda **Article**), prikazovanju vsebine košarice na zaslon itd.

Vsebina spletne košarice, implementirane na takšen način, bi se ob ponovnem nalaganju strani izgubila. Zato smo na strežniški strani implementirali skripto, ki v sejnih spremenljivkah hrani celotno vsebino košarice. Naloga razreda **Cart** je, da skrbi za komunikacijo s strežnikom in mu ob vsaki spremembi košarice pošlje osvežene podatke ter pridobi podatke s strežnika, ko se stran ponovno naloži.



Slika 3.13: Prikaz košarice.

3.3.3.4 Pregled naročila

Ko v košarici pritisnemo na gumb '*Nadaljuj z naročilom*' nas sistem preusmeri na pregled naročila, kjer nastavimo še zadnje podrobnosti.

Pregled naročila

Naslov za dostavo
Levčeva ulica 17, 1234 Mengeš

[Ste pozabili kaj naročiti?](#)

Hitra sedmica

1	Margerita, velika	6.00
1	Kmečka, mala	6.50
		12.50 EUR

Komentar za restavracijo

Način plačila: Gotovina

Način prevzema: Dostava (takoj)

Skupaj (z DDV) 12,50 EUR

Nadaljuj z naročilom

Slika 3.14: Prikaz pregleda naročila.

Na levi strani so navedeni podatki o naslovu naročnika, na desni pa pregled košarice.

S spustnega seznama izberemo naslov za dostavo ali pa napišemo novega s pomočjo obrazca.

Na desni spreminjamo količino posameznega artikla ali ga celo odstranimo iz košarice. Za vsako restavracijo izberemo način prevzema in način plačila, seveda, če restavracija nudi več možnosti. Restavraciji pa lahko napišemo tudi sporočilo.

Če restavracija ne dostavlja v izbrani kraj, se izpiše ustrezno obvestilo, sistem pa onemogoči nadaljevanje naročila. To se zgodi tudi v primeru, če želimo naročiti hrano iz restavracije izven odpiralnega časa.

3.3.4 Sistem za predlaganje pic

Sistem za predlaganje pic temelji predvsem na iskanju sorodnih pic. Ko smo izdelali mehanizem za razvrščanje pic v kategorije, je to opravilo bistveno enostavnejše, saj imamo pice že nekako razporejene po skupinah.

Naš predlagalni sistem deluje na štiri načine, in sicer:

1. Predlaganje pic za uporabnike brez zgodovine

Če uporabnik prek našega sistema še ni naročil pice, potem predlagamo najpogostejše naročene pice v izbrani restavraciji.



Slika 3.15: Prikaz predlaganja najpogostejše naročane pice.

2. Predlaganje pic na podlagi uporabnikove zgodovine

Če je uporabnik že naročal pice prek našega sistema, mu predlagamo sorodne pice (npr. zelenjavne pice). To storimo tako, da pridobimo podatke, iz katere kategorije oz. katerih kategorij so bile najpogostejše uporabnikove naročene pice. Ko imamo znane kategorije, moramo iz posamezne restavracije samo še izbrskati pice, ki ustrezajo tem kategorijam.

3. Predlaganje pic na podlagi uporabnikove zgodovine in naročil drugih uporabnikov

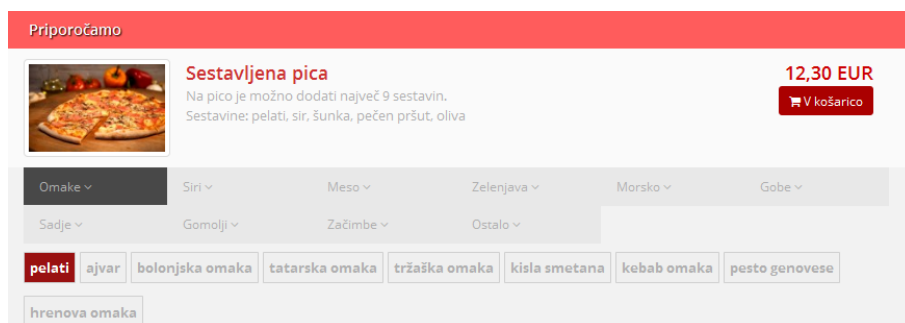
Sistem predlaga pico na podlagi naročil drugih uporabnikov s podobnim okusom. To storimo tako, da pregledamo vse uporabnike, ki so v določeni restavraciji naročali največkrat pice iz tiste kategorije kot uporabnik, za katerega predlagamo pico (npr. pikantne mesne pice). Nato od teh uporabnikov pridobimo seznam naročenih pic, ki ne pripadajo uporabnikovi najljubši kategoriji (pikantnim mesnim picam), in

eno izberemo. Tako dobimo tudi nekoliko manj sorodne pice, ki jih uporabnik običajno naroča.

4. Predlaganje sestavljene pice

Če restavracija ponuja možnost sestavljanja pic, uporabniku predlagamo sestavine za sestavljeno pico. Sestavine določimo glede na sestavine pic, ki jih je uporabnik že naročal. Uporabnik lahko predlagane sestavine poljubno odstrani ali dodaja nove.

Sestavine predlagamo tako, da iz vsake kategorije sestavin izberemo tisto sestavino, ki se je najpogosteje znašla na picah, ki jih je uporabnik naročal. Če restavracija ponuja večje število sestavin na sestavljeni pici, potem naključno izberemo še ostale sestavine.



Slika 3.16: Prikaz predlaganja vnaprej sestavljene pice.

Sistem naključno izbira med načini za predlaganje pice, ki izpolnjujejo potrebne pogoje. Pri prvem načinu mora restavracija že imeti vsaj eno naročeno pico, da izpolni potreben pogoj. Pri drugem in četrtem mora imeti poleg tega še uporabnik svojo zgodovino naročil. Pri tretjem načinu pa moramo imeti skupino uporabnikov s svojo zgodovino.

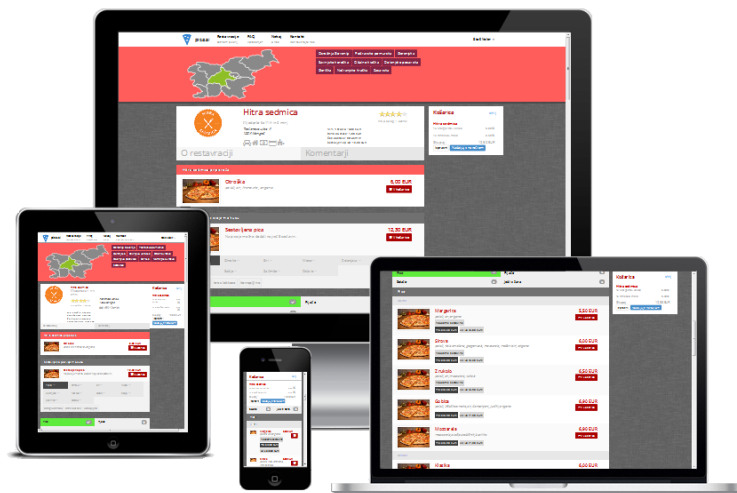
Sistem uporabnike klasificira v določene skupine, tako da se nakupi ne odražajo pri predlaganju samo v eni restavraciji, ampak v vseh, četudi tam uporabnik še ni naročal.

3.3.5 Odzivna spletna stran

Za izdelavo odzivne spletne strani smo se odločili zaradi vse večjega porasta tabličnih računalnikov, mobilnih telefonov in nasploh sodobnih naprav z različnimi velikostmi zaslonov, s katerimi lahko dostopamo do spleta.

Med izdelovanjem odzivne spletne strani moramo pri postavitvi elementov razmišljati tudi, kako se bodo elementi prikazovali na manjših zaslonih. Elementom moramo zato podati ustrezne lastnosti. Ker je definiranje teh lastnosti lahko zelo zamudno, smo uporabili ogrodje Twitter Bootstrap, ki ima že definirane razrede CSS, ki jih lahko uporabljamo. Poleg tega pa ima precej uporabnih sestavljenih komponent, kot je npr. meni, ki smo ga tudi s pridom uporabili.

Največjo dilemo smo imeli pri postavitvi košarice na manjših velikosti zaslonov. Odločili smo se, da jo postavimo tako, da je ves čas vidna na vrhu zaslona in da je razpotegnjena čez celo širino. Problem nastane, če je v košarici veliko različnih artiklov in se košarica razpotegne čez vso višino zaslona, tako da nam je onemogočeno brskanje po strani. Zato smo košarici dodali možnost zapiranja in odpiranja.



Slika 3.17: Prikaz spletne strani na različnih napravah.

Poglavje 4

Analiza in odziv uporabnikov

S primerjavo s podobnimi že obstoječimi aplikacijami za naročanje hrane smo ugotovili glavne prednosti in slabosti izdelane aplikacije. Ocenili smo pravilnost razvrščanja pic. Za mnenje pa smo povprašali uporabnike, ki so aplikacijo testirali.

4.1 Primerjava z obstoječimi aplikacijami

Danes obstaja precej spletnih aplikacij, prek katerih lahko naročamo hrano. Tudi v Sloveniji imajo restavracije možnost prodaje s pomočjo tovrstnih aplikacij. Našo aplikacijo smo primerjali še z dvema podobnima, ki uspešno delujeta na slovenskem trgu, in sicer Ehrana ter Halohrana.

Spletne aplikacije smo med seboj primerjali glede na *iskanje restavracij in naročanje, pristope za čimbolj učinkovito prodajo ter prikaz strani na različnih napravah*:

- **Iskanje restavracij**

Vsem trem je skupno enostavno iskanje restavracij s pomočjo različnih filtrov. Pri naši aplikaciji in portalu Ehrana si lahko pri iskanju pomagamo še z zemljevidom Slovenije. Za razliko od naše aplikacije lahko pri ostalih dveh pri iskanju uporabimo še filter za izbiro vrste hrane oz.

tip kuhinje. V našo aplikacijo tega nismo vključili, saj se osredotočamo predvsem na prodajo samo ene vrste hrane.

- **Naročanje in dodajanje artiklov v košarico**

Naročanje in dodajanje artiklov v košarico je pri vsaki urejeno malce drugače. Ehrana omogoča naknadno spreminjanje dodatkov posameznim jedem, tudi, ko so že v košarici, medtem ko pri naši aplikaciji in portalu Halohrana lahko naknadno spreminjamo samo količino artiklov, sicer moramo artikel odstraniti iz košarice in dodati novega s spremenjenimi dodatki. Zato pa je prednost naše aplikacije in Halohrane naročanje hrane iz več restavracij hkrati in možnost naročila z dostavo ali prevzemom ob točno določenem času.

- **Pristopi za čimbolj učinkovito prodajo**

Namen vsake trgovine je čimbolj učinkovita prodaja, zato je zelo priporočljivo, da ima vgrajene mehanizme, ki uporabnika čimbolj pritegne h končnemu nakupu. Od različnih popustov in akcijskih cen ter predlaganja artiklov do zbiranja točk zvestobe. Portal Ehrana ima mehanizem za določanje različnih popustov, medtem ko Halohrana ponuja možnost zbiranja točk zvestobe, ki jih lahko kasneje kupec unovči ob naslednjem nakupu. Poleg določanja akcijskih cen artiklom je glavna prednost naše aplikacije sestavljanje lastne pice in sistem za predlaganje pic glede na uporabnikovo zgodovino naročanja.

- **Prikaz strani na različnih napravah**

Prednost naše strani je, da se oblika prilagaja velikosti zaslona naprave, tako da je enako dobro vidna tako na namiznih kot na tabličnih računalnikih in mobilnih telefonih. Pri ostalih dveh se oblika ne prilagaja velikosti zaslona, ima pa Ehrana izdelano posebno stran, prilagojeno samo za mobilne telefone.

Menimo, da smo izdelali celovit sistem za naročanje hrane, ki se od ostalih sorodnih aplikacij razlikuje v tem, da se naša posebej osredotoča na prodajo

ene vrste hrane in ima v te namene izdelane mehanizme za čimbolj učinkovito prodajo.

Podoben sistem bi lahko uporabili za predlaganje storitev ali izdelkov izven kulinarike. Na primer, predlaganje turističnih destinacij na podlagi preteklih potovanj, le da bi namesto sestavin uporabili tipe nastanitve (hotel, apartma, kamp . . .), destinacije (morje, podeželje, hribi, regije, kraji . . .) itd.

4.2 Pravilnost samodejnega razvrščanja pic

Samodejno razvrščanje pic v ustrezne kategorije je ključni element pri sistemu za predlaganje pic, zato smo želeli, da bi razvrščanje delovalo čimbolj realno.

Za pregled pravilnosti delovanja smo ustvarili šest glavnih kategorij: *sirove*, *mesne*, *zelenjavne*, *vegetarijanske*, *morske* in *sadne*. Vsaki smo določili pravila za razvrščanje, nato pa vnesli testne pice. Ta postopek smo nekajkrat ponovili, saj smo s spremembami pravil želeli doseči čim boljše razvrščanje. Glavni problem pri tem je bilo določanje mej med posameznimi kategorijami, saj gre tu lahko pogosto za stvar osebne interpretacije. Na primer, nekateri bi pico z dvema različnima vrstama sira in eno vrsto zelenjave uvrstili med sirove, drugi med zelenjavne, tretji pa med vegetarijanske. Zato smo morali sprejeti kompromis in določiti meje tako, da bodo v splošnem z razvrščanjem na koncu večinoma vsi zadovoljni.

4.3 Odziv uporabnikov

Aplikacijo so testirali trije uporabniki, dva na namiznem računalniku, eden pa na mobilnem telefonu. Preizkusili so delovanje tako prednjega kot zalednega dela.

- **Preglednost in uporaba spletne strani**

Vsi trije so v splošnem pohvalili preglednost spletne strani, enostavnost iskanja restavracij in naročanja. Testnega uporabnika na mobilnem telefonu je malce zmotila le nepreglednost pri izbiri dodatkov posameznim jedem. Predlagal je tudi iskanje restavracij, ki so najbližje trenutni lokaciji.

- **Predlagalni sistem in samodejno razvrščanje pic**

Vsi trije so bili navdušeni nad predlagalnim sistemom, še posebej za predlagane sestavljene pice.

Pri vnašanju pic so pohvalili samodejno razvrščanje v ustrezne kategorije, čeprav je nekdo predlagal, da bi samodejno razvrščanje uporabil za interno uporabo za predlaganje pic, drugo razvrščanje pa bi bilo vidno na meniju in bi ga lahko ob vnosu nove pice lastnik restavracije ali administrator nastavil ročno.

- **Prejemanje naročil za restavracije**

Všeč jim je bil način prejetanja naročil za lastnike restavracij in njihove zaposlene, pri tem so predlagali le še glasovno opozorilo, ki naznanja novo naročilo.

Poglavje 5

Sklepne ugotovitve

V prvem delu smo se osredotočili na pregled uporabljenih tehnologij in orodij, ki smo jih uporabili za izdelavo spletne aplikacije. V drugem delu pa je prikazan razvoj spletne aplikacije od analize zahtev do implementacije. Podrobneje so opisani glavni deli aplikacije. Nekaj več časa smo na začetku namenili načrtovanju, kar se nam je kasneje obrestovalo, saj je implementacija tako potekala hitreje kot bi sicer. Aplikacijo smo med izdelavo sproti testirali v različnih spletnih brskalnikih.

Med izdelovanjem smo pridobili ogromno idej, kako bi aplikacijo lahko v prihodnosti še izboljšali in obogatili. Pri vsaki piceriji bi lahko izdelali še poseben filter pri meniju, kjer bi iskali pice s točno določenimi sestavinami. Sestavljanje pice bi lahko obogatili z grafičnim prikazom, kjer bi bile vidne izbrane sestavine. Restavracijam bi lahko omogočili dodajanje posebnih akcij in popustov, možnost dodajanja obrazca za rezervacije mize v restavraciji, kreiranje fotogalerije in možnost generiranja poročil. Za dostavljavce hrane bi lahko izdelali poseben mehanizem, ki bi jim glede na naročila in lokacije izrisal najboljšo pot na zemljevidu. Z izvedbo uporabniškega testiranja, bi pridobili različne odzive uporabnikov in na podlagi teh bi lahko še izboljšali uporabniško izkušnjo.

Če razmišljamo še malce širše, bi lahko, čeprav je aplikacija izdelana z odzivnim pogledom, izdelali še mobilno aplikacijo za uporabnike. Posebno

aplikacijo bi lahko izdelali tudi za zaposlene, tako da bi jih že mobilni telefon lahko opozarjal na nova naročila.

Kljub temu da je še veliko prostora za izboljšave, v splošnem ocenjujemo, da je izdelana spletna aplikacija za naročanje hrane v okviru diplomskega dela realizirana v skladu z zahtevami, pričakovanji in cilji.

Slike

3.1	Diagram primerov uporabe za obiskovalca. Prikaže aktivnosti prednjega dela aplikacije.	10
3.2	Diagram primerov uporabe za uporabnika. Prikaže aktivnosti prednjega dela aplikacije.	11
3.3	Diagram primerov uporabe za zaposlenega. Prikaže aktivnosti zalednega dela aplikacije.	12
3.4	Diagram primerov uporabe za lastnika restavracije. Prikaže aktivnosti zalednega dela aplikacije.	13
3.5	Diagram primerov uporabe za administratorja. Prikaže aktivnosti zalednega dela aplikacije.	14
3.6	Prikaz urejanja podatkov posamezne restavracije v zalednem sistemu za administratorje.	20
3.7	Prikaz pregledovanja aktivnih naročil za lastnike restavracij in njihove zaposlene.	21
3.8	Prikaz določanja deležev sestavin za kategorijo sirovih pic. . .	22
3.9	Prikaz prednje strani.	27
3.10	Prikaz posamezne restavracije.	28
3.11	Prikaz menija pri posamezni restavraciji.	29
3.12	Prikaz filtra za hitrejše iskanje restavracij.	30
3.13	Prikaz košarice.	30
3.14	Prikaz pregleda naročila.	31
3.15	Prikaz predlaganja najpogostejše naročane pice.	32
3.16	Prikaz predlaganja vnaprej sestavljene pice.	33

3.17 Prikaz spletne strani na različnih napravah.	34
---	----

Literatura

- [1] (2015) HTML, dostopno na:
<http://en.wikipedia.org/wiki/HTML>
- [2] (2015) CSS, dostopno na:
http://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [3] (2015) CSS, dostopno na:
<http://sl.wikipedia.org/wiki/CSS>
- [4] (2015) PHP, dostopno na:
<http://en.wikipedia.org/wiki/PHP>
- [5] R. Lerdorf, P. Macintyre, K. Tatroe.
“Programming PHP, 3rd Edition”, 2013
- [6] (2015) PHP just grows & grows, dostopno na:
<http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>
- [7] (2015) Vendor Specific Database Extensions, dostopno na:
<http://php.net/manual/en/refs.database.vendors.php>
- [8] (2015) MySQL, dostopno na:
<http://en.wikipedia.org/wiki/MySQL>
- [9] Paul Dubois. “MySQL, 4th Edition”, 2008

- [10] (2015) DB-Engines Ranking, dostopno na:
<http://db-engines.com/en/ranking>
- [11] (2015) JavaScript, dostopno na:
<http://en.wikipedia.org/wiki/JavaScript>
- [12] (2015) JavaScript, dostopno na:
<http://sl.wikipedia.org/wiki/JavaScript>
- [13] (2015) AJAX, dostopno na:
[http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- [14] Anthony T Holdener III. "Ajax: The Definitive Guide", 2008
- [15] (2015) jQuery, dostopno na:
<http://www.jquery.com>
- [16] (2015) Usage of JavaScript libraries for websites, dostopno na:
http://www.w3techs.com/technologies/overview/javascript_library/all
- [17] (2015) jQuery UI, dostopno na:
<http://www.jqueryui.com>
- [18] (2015) Twitter Bootstrap, dostopno na:
<http://www.getbootstrap.com>
- [19] (2015) Select2, dostopno na:
<http://ivaynberg.github.io/select2>
- [20] (2015) Star Rating, dostopno na:
<http://plugins.krajee.com/star-rating>
- [21] (2015) FancyBox, dostopno na:
<http://fancybox.net>
- [22] (2015) Bootstrap WYSIHTML5, dostopno na:
<https://github.com/xing/wysihtml5>

- [23] (2015) Font Awesome, dostopno na:
<http://fontawesome.github.io/Font-Awesome>
- [24] (2015) RaphaëlJS, dostopno na:
<http://raphaeljs.com>
- [25] (2015) Notepad++, dostopno na:
<http://notepad-plus-plus.org>
- [26] (2015) MySQL Workbench, dostopno na:
<http://www.mysql.com/products/workbench/>